

## Содержание

Введение.....	6
1. Анализ технического задания .....	7
2. Проектирование системы .....	9
3. Реализация системы .....	13
4. Тестирование системы.....	15
Заключение .....	18
Список используемой литературы .....	19
Приложение А. Проект системы (диаграммы)	<b>Ошибка! Закладка не определена.</b>
Приложение Б. Исходный текст программы	<b>Ошибка! Закладка не определена.</b>

					<b>МИВУ.09.03.04-18.000 ПЗ</b>		
<b>Изм.</b>	<b>Лист</b>	<b>№ докум.</b>	<b>Подпись</b>	<b>Дата</b>	<b>ПОЯСНИТЕЛЬНАЯ ЗАПИСКА</b>		
Разраб.		Сидоров А.С.					
Провер.		Привезенцев Д.Г.					
Реценз.							
Н. Контр.							
Утверд.							
						<b>Лит.</b>	<b>Лист</b>
							<b>Листов</b>

## Введение

В настоящее время мир переживает информационную революцию, вызванную широким внедрением в жизнь общества Интернета и Всемирной паутины, которая информационно связывает все сферы деятельности, все организации и конкретных людей. Суть этой революции заключается в интеграции в едином мировом информационном пространстве программно-аппаратных средств, средств связи и телекоммуникаций, информационных ресурсов и накопленных знаний в единую информационную и коммуникационную инфраструктуру.

Решающее значение для информационных стратегий организаций является присутствие в Интернете и создание распределенных информационных систем, обеспечивающих взаимодействие с заказчиками и поставщиками, маркетинг и многие другие виды деятельности. Все это вызывает повышенный интерес к технологиям создания инфокоммуникационных систем, основанных на распределенных вычислениях.

Главная задача курсовой работы – это изучить протокол передачи информации TCP и применить эти знания в практической сфере.

Протокол TCP (Transmission Control Protocol, Протокол контроля передачи) обеспечивает сквозную доставку данных между прикладными процессами, запущенными на узлах, взаимодействующих по сети.

TCP - надежный байт-ориентированный протокол с установлением соединения. TCP находится на транспортном уровне стека TCP/IP, между протоколом IP и собственно приложением. Протокол IP занимается пересылкой дейтаграмм по сети, никак не гарантируя доставку, целостность, порядок прибытия информации и готовность получателя к приему данных; все эти задачи возложены на протокол TCP.

Цель курсовой работы является разработка приложения для удаленного управления компьютером.

					МИВУ.09.03.04-18.000 ПЗ	Лист
						6
Изм.	Лист	№ докум.	Подпись	Дата		

## 1. Анализ технического задания

Целью курсовой работы является разработка сетевого приложения.

Исходя из поставленной цели, можно выделить следующие задачи курсовой работы:

- Спроектировать систему
- Разработать и реализовать необходимые функции
- Протестировать программу на выявление ошибок

Система состоит из двух частей:

- Клиентская часть
- Серверная часть

Программа клиентской части должна содержать следующие функциональные возможности:

- Осуществлять подключение по идентификатору

Идентификатор генерируется в клиентском приложении при запуске;

- Осуществлять передачу данных (команд и скриншотов)

Ведущий подключается к компьютеру ведомого. Далее начинается пересылка скриншотов от управляемого компьютера пользователя управляющему пользователю. Последний пересылает действия и команды в обратном направлении, где происходит их исполнение.

Программа серверной части должна содержать следующие функциональные возможности:

- Принимать и отправлять команды

Сервер считывает команды от пользователя, обрабатывает их и, при необходимости, посылает ответ;

- Ведение списка пользователей;

Сервер создает список пользовательских идентификаторов и адресов подключения, при появлении нового подключения к сети сервера.

Система реализуется с помощью протокола ТСР.

					МИВУ.09.03.04-18.000 ПЗ	Лист
						7
Изм.	Лист	№ докум.	Подпись	Дата		

Модуль ТСР выполняет передачу непрерывных потоков данных между своими клиентами в обоих направлениях. Клиентами ТСР являются прикладные процессы, вызывающие модуль ТСР при необходимости получить или отправить данные процессу-клиенту на другом узле.

Разработанная программа клиентской части представляет собой десктопное приложение с графическим интерфейсом. Серверная часть представлена в виде консольного приложения.

Сервер имеет ограниченное подключение пользователей, заданное администратором. Количество согласно заданию – до 8 подключений.

При инициализации удаленного управления ведущим посылается запрос ведомому, которому необходимо подтвердить (или отклонить) попытку подключения к компьютеру.

Среда разработки сетевого приложения Visual Studio 2019, язык программирования С#.

					МИВУ.09.03.04-18.000 ПЗ	Лист
						8
Изм.	Лист	№ докум.	Подпись	Дата		

## 2. Проектирование системы

При проектировании системы необходимо рассмотреть концепцию работы всей системы в целом. И уже на основе этого анализа построить функциональную модель.

Для этого вначале строится модель в форме так называемой диаграммы вариантов использования, которая описывает функциональное назначение системы или, другими словами, то, что система будет делать в процессе своего функционирования. Диаграмма вариантов использования является исходным концептуальным представлением или концептуальной моделью системы в процессе ее проектирования и разработки. (Приложение А. Рисунок 8)

Актерами в диаграмме являются Пользователь и Администратор. Пользователь, на которого ориентирована программа может присоединиться к серверу путём ввода IP-адреса сервера. Далее ведущий пользователь может управлять ведомым компьютером после ввода уникального идентификатора ведомого пользователя, последний должен разрешить это подключение. Администратор же отвечает за запуск сервера или его остановку, так же ему доступны команды сервера.

Наиболее удобным языком моделирования, дающую адекватное отображение предметной области и всех участников бизнес-процессов организации, является нотация IDEF0. Под моделью в IDEF0 понимается описание системы (текстовое и графическое), которое должно дать ответ на некоторые заранее поставленные вопросы. При этом технология IDEF0 реализует методологию функционального описания систем в целом как множество взаимозависимых действий или функций.

Диаграмма клиентской части (Приложение А. Рисунок 9) показывает необходимые входные данные, лица, которые участвуют в выполнении задач на клиентской части, правила выполнения тех или иных задач, а также выходные данные.

					МИВУ.09.03.04-18.000 ПЗ	Лист
						9
Изм.	Лист	№ докум.	Подпись	Дата		

Декомпозиция диаграммы клиентской части (Приложение А. Рисунок 10) показывает подробное выполнение программы на клиентской стороне.

Для подключения к серверу через протокол ТСР/ІР пользователю необходимо ввести ІР адрес сервера и порт, далее нажать кнопку «Соединиться с сервером». После установке соединения приложение ожидает команд от сервера, в случае которых происходит инициализация передачи. Так же пользователь может инициализировать подключение к другому пользователю посредством указания уникального идентификатора и нажатием кнопки «Подключиться». После соединения клиентского приложения с серверной частью, разрешена команда инициализации передачи или управления.

Декомпозиция функционального блока «Инициализация управления» диаграммы клиентской части (Приложение А. Рисунок 11) показывает подробное выполнение этого блока на клиентской стороне.

Для инициализации управления пользователь вводит уникальный идентификатор пользователя, которым хочет управлять и нажимает кнопку подключиться. Далее происходит отправка команды серверу, от которого ожидается ответ. При поступлении последнего, выводится окно сообщения о результате подключения. В случае успеха устанавливается соединение с удаленным пользователем и открывается форма для удаленного управления компьютером. В нее выводятся скриншоты, присланные пользователем, и считываются команды для передачи на удаленный компьютер пользователя.

Декомпозиция функционального блока «Инициализация передачи» диаграммы клиентской части (Приложение А. Рисунок 12) показывает подробное выполнение этого блока на клиентской стороне.

В случае запроса на подключения пользователю выдается запрос на разрешения подключения, после которого ответ пользователя будет отправлен в виде команды серверу. При положительном ответе пользователя ожидается соединение с ведущим пользователем, при установлении такого приложение

отправляет скриншоты своего экрана и исполняет пришедшее управляющие команды от ведущего пользователя.

Диаграмма серверной части (Приложение А. Рисунок 13) показывает необходимые входные данные, лица, которые участвуют в выполнении задач на серверной части, правила выполнения тех или иных задач, а также выходные данные.

Декомпозиция диаграммы серверной части (Приложение А. Рисунок 14) показывает подробное выполнение программы на серверной стороне.

Для запуска сервера через протокол TCP/IP администратору необходимо ввести порт, который будет использоваться для подключения клиентских приложений. Далее сервер переходит в режим прослушки команд от пользователей. При установлении соединения с сервером, происходит регистрация пользователя в списке, с указанием его идентификатора и удаленного адреса подключения. Сервер обрабатывает пришедшие некоторые команды, путем пересылки из заданному в команде идентификатору. Другие некоторые команды вынуждают сервер удалить из списка пользователей, указанных в команде пользователей. Администратор так же может запросить список находящихся в сети пользователей.

Диаграммы потоков данных (Data flow diagram, DFD) используются для описания документооборота и обработки информации. Подобно IDEF0, DFD представляет моделируемую систему как сеть связанных между собой работ. Их можно использовать как дополнение к модели IDEF0 для более наглядного отображения текущих операций документооборота в корпоративных системах обработки информации.

Главная цель DFD - показать, как каждая работа преобразует свои входные данные в выходные, а также выявить отношения между этими работами. Любая DFD-диаграмма может содержать работы, внешние сущности, стрелки (потоки данных) и хранилища данных.

На диаграмме DFD (Приложение А. Рисунок 15) представлено что сервер использует список пользователей, в котором содержатся все пользователи, подключенные к серверу. Выполняет пересылку команд между удаленными пользователями, по пришедшей команде пользователя отправляет ее, используя команду сервера.

Диаграмма состояний по существу является графом специального вида. Вершинами этого графа являются состояния и некоторые другие типы элементов, которые изображаются соответствующими графическими символами. Дуги графа служат для обозначения переходов из состояния в состояние.

На диаграмме состояния клиентской части приложения (Приложение А. Рисунок 16) и серверной части приложения (Приложение А. Рисунок 17) показаны активность пользователя и сервера при происхождении тех или иных событий.

Диаграммы последовательности являются видом диаграмм взаимодействия языка UML, которые описывают отношения объектов в различных условиях. Условия взаимодействия задаются сценарием, полученным на этапе разработки диаграмм вариантов использования.

На диаграмме последовательности (Приложение А. Рисунок 18) представлены поэтапные взаимодействия пользователей с приложением, администратора с сервером, приложениями пользователей с сервером.

Диаграмма размещения отражает физические взаимосвязи между программными и аппаратными компонентами системы. Она является хорошим средством для того, чтобы показать маршруты перемещения объектов и компонентов в распределенной системе.

Каждый узел на диаграмме размещения представляет собой некоторый тип вычислительного устройства, в большинстве случаев – часть аппаратуры.

На диаграмме размещения (Приложение А. Рисунок 19) представлены рабочие части для инициализации удаленного управления между клиентами. Взаимодействие между клиентами и сервером осуществляется путем отправки команд для запусков процедур и обработки данных. При инициализации управления или передачи соединения устанавливается между клиентами.



### 3. Реализация системы

Разрабатываемая система состоит из двух частей:

- Клиентской
- Серверной

Клиентская часть представлена в виде формы для удобства использования.

Рисунок 1 – Форма клиентской части

Клиентская форма состоит из:

- 1) Поля ввода адреса сервера

Пользователь вводит уже известный адрес сервера и порта, для подключения к нему.

- 2) Кнопка подключения клиентской части к серверу

По нажатию происходит попытка подключения к серверу, после успешного подключения клиентская часть отправит сообщение серверу для регистрации в списке пользователей. Сервер ответит о успехе подключения. В положительном случае кнопка перестанет отвечать на действия пользователя.

- 3) Поля ввода идентификатора клиента

Пользователь вводит уникальный идентификатор пользователя (клиента), для установления соединения и дальнейшего удаленного управления его компьютером.

#### 4) Кнопка подключиться

По нажатию происходит отправка сообщения серверу об установлении подключения с компьютером, указанном по уникальному номеру пользователя. Сервер вернет ответ, в положительном случае будет открыта форма удаленного управления пользователем (клиентом). На форме отображаются скриншоты, присланные удаленным пользователем. Также ведется считывание команд, введенных локальным пользователем, и пересылка удаленному пользователю.

Серверная часть представлена в виде консоли.

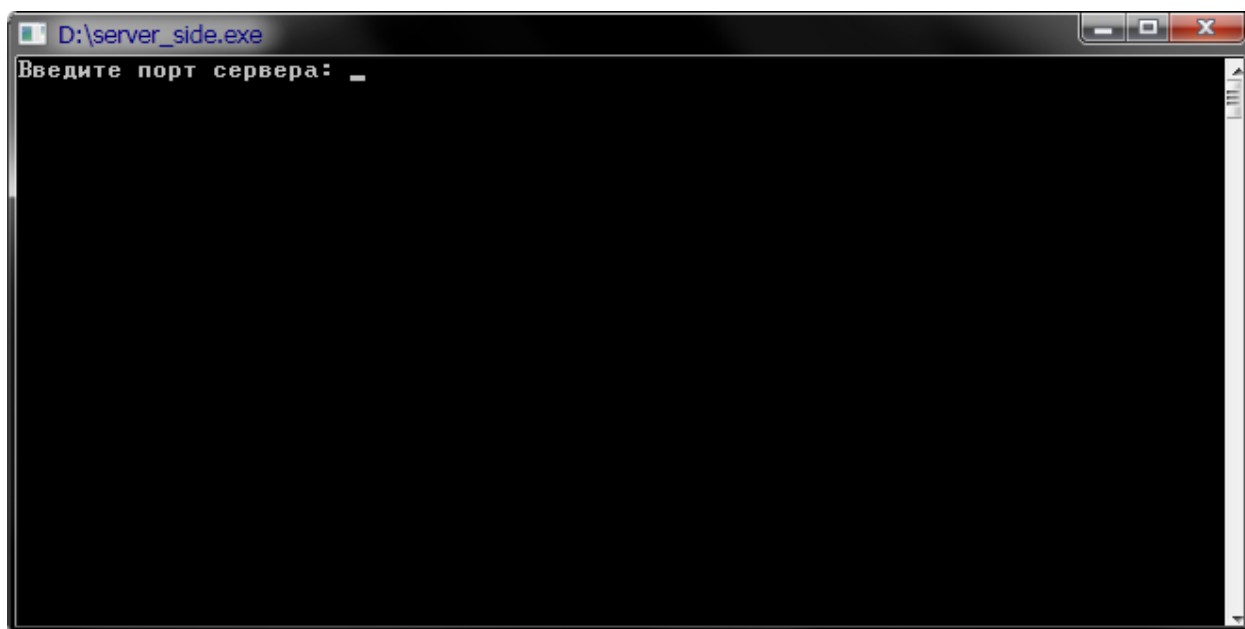


Рисунок 2 – Форма серверной части

Для запуска сервера необходимо ввести порт, по которому будут подключаться пользователи (клиенты), далее сервер будет осуществлять прослушку входящих соединений и их обработку. Сервер выводит на консоли информацию о проведенных операциях. Так же администратор может в любой момент ввести команду в консоль сервера, последний отработает её.

#### 4. Тестирование системы

Одним из важнейших этапов создания приложения является его тестирование и отладка. Тестирование позволяет выявить скрытые и явные недостатки программы, либо убедиться в ее пригодности для применения. Обнаруженные недостатки устраняются в ходе отладки.

Целью тестирования является проверка работоспособности программы, правильности выполнения всех функций, описанных выше, а также правильности обработки всех исключений, возникающих в ходе работы программы.

##### 1. Запуск сервера

Для начала нужно ввести порт, далее сервер начнет работу.

Сервер запустился корректно, ошибок не обнаружено;

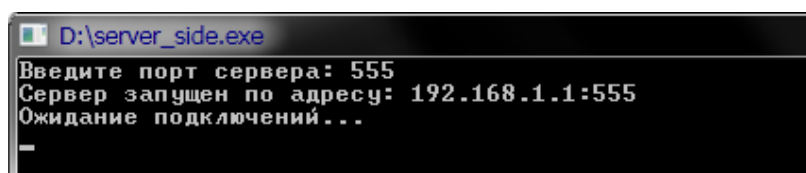


Рисунок 3. Окно консоли с ответом о успешном запуске

##### 2. Подключение пользователя к серверу

Запуск клиентского приложения на компьютере пользователя и установление соединения с сервером.

Пользователь подключен, сообщение в консоли отображено.

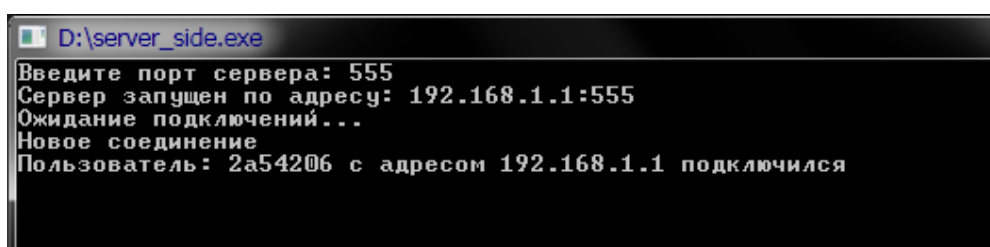


Рисунок 4. Окно консоли с ответом о подключении пользователя

### 3. Ввод неправильной команды

Введем некоторые команды в консоль сервера;

Команда тест исполнена. Другие команды на сервере не найдены.

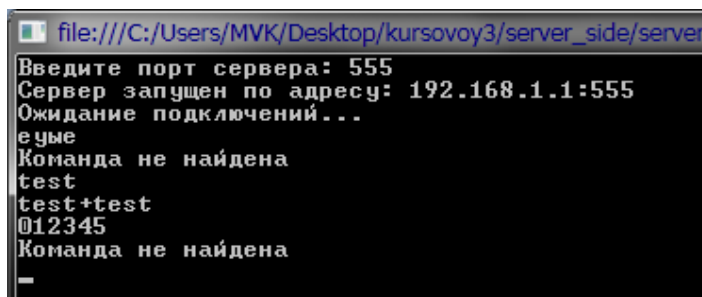


Рисунок 5. Окно консоли с ответом о некорректной команде

### 4. Отправка запроса при подключении.

Выполняем подключение для удаленного управления с локального компьютера на удаленный, зная его идентификатор.

На удалённом компьютере пользователя отображено сообщение о подключении к нему ведущего пользователя.

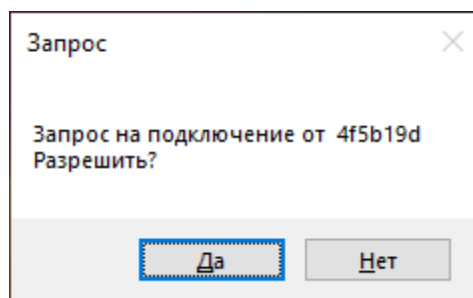


Рисунок 6. Окно запроса пользователя на управление компьютером

### 5. Окно удаленного управления

Выполним соединение локального и удаленного компьютера, разрешим удаленное управление и посмотрим на форму.

Программа успешно вывела форму удалённого управления компьютером пользователя.

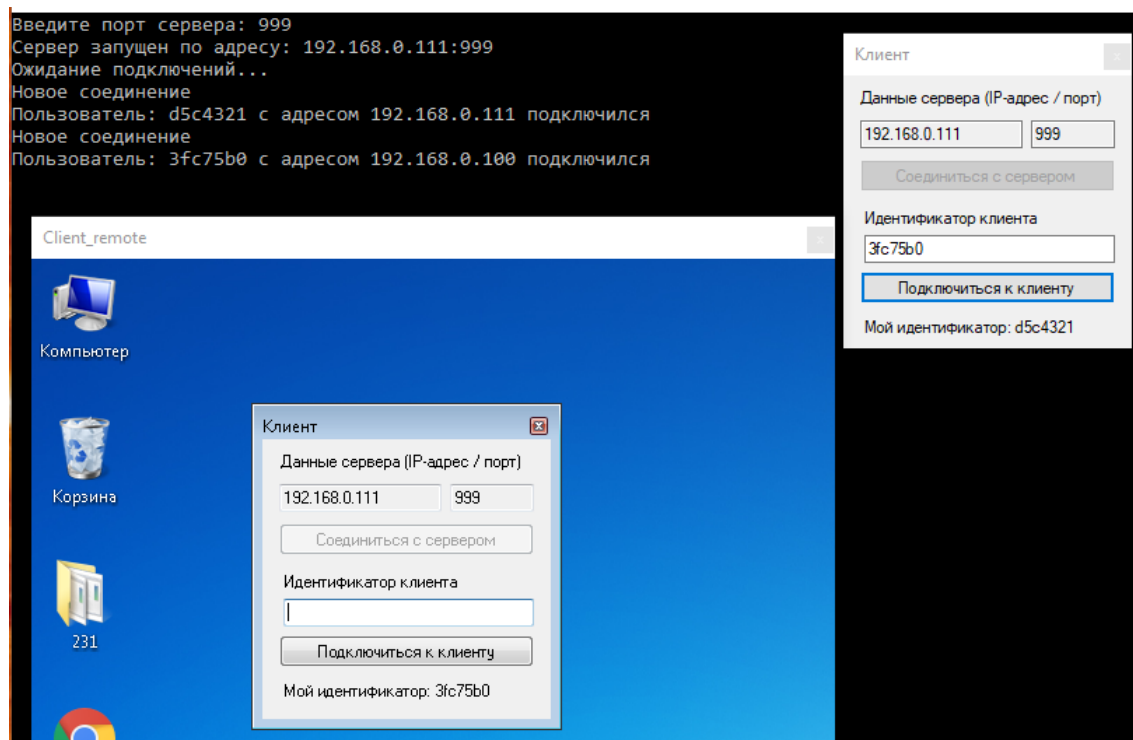


Рисунок 7. Форма демонстрации экрана удаленного пользователя

В результате тестирования программы, выявленные ошибки были устранены. В текущей версии, недостатков, влияющих на корректную работу приложения, не выявлено.

Данная программа тестировалась на компьютере следующей конфигурации:

- Тип ЦП: intel core i3-7100, 3.9 ГГц
- ОЗУ: 8GB – DDR4
- Операционная система Microsoft Windows 10 (x64)

## Заключение

					МИВУ.09.03.04-18.000 ПЗ	Лист
						18
Изм.	Лист	№ докум.	Подпись	Дата		

Список используемой литературы